**A Project Report**
**On**

**IMPLEMENTATION OF VOICE AND GESTURE CONTROL APPLICATION TO SMART ADAPTIVE VEHICLE**

Submitted for partial fulfillment of the requirements for the award of the degree of

**BACHELOR OF ENGINEERING**
**IN**
**ELECTRICAL AND ELECTRONICS ENGINEERING**

**BY**


| | |
|---|---|
| **K.V.N. SHARANYA** | **(1608-19-734-003)** |
| **M.ANIL** | **(1608-19-734-018)** |
| **J. AKSHITH KUMAR** | **(1608-19-734-029)** |


Under the guidance of
**Mr. GVSSN SRIRAMA SARMA, M.Tech, (Ph.D)**
Assistant Professor


**Department of Electrical and Electronics Engineering**
**MATRUSRI ENGINEERING COLLEGE,**
**(Sponsored by Matrusri Education society, Estd1980)**
**(Approved by AICTE, Affiliated to Osmania University)**
**#16-1-486, Saidabad, Hyderabad, Telangana-500 059**
**www.matrusri.edu.in**

**2022-2023**

**Matrusri Engineering College**
(Sponsored by Matrusri Education Society, Estd.1980)
(Approved by AICTE & Affiliated to Osmania University)
# 16-1-486, Saidabad, Hyderabad-500059. Ph: 040-24072764
(ISO 9001:2015 Certified, Accredited by NBA)

NATIONAL BOARD OF ACCREDITATION

A+ GRADE Accredited by NAAC

## Department of Electrical and Electronics Engineering

## CERTIFICATE

This is to certify that the project work entitled "**IMPLEMENTATION OF VOICE AND GESTURE CONTROL APPLICATION TO SMART ADAPTIVE VEHICLE**" is a bonafide work carried out by **K.V.N. SHARANYA(**1608-19-734-003**), M. ANIL(**1608-19-734-018**),J.AKSHITH KUMAR (**1608-19-734-029) in partial fulfillment for the award of the Degree of Bachelor of Engineering in Electrical and Electronics of Matrusri Engineering college, Hyderabad during 2019-23, is a record of bonafide work carried out under our guidance and supervision.

The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.

**Signature of Supervisor**
Mr.GVSSNSriramaSarma
Assistant Professor

**Signature of Head of Department**
Dr. G. Ravindranath
Professor&HOD, EEEDepaartment

**Signature of External Examiner**

Matrusri Engineering College
(Sponsored by Matrusri Education Society, Estd.1980)
(Approved by AICTE & Affiliated to Osmania University)
# 16-1-486, Saidabad, Hyderabad-500059. Ph: 040-24072764
(ISO 9001:2015 Certified, Accredited by NBA)

# DECLARATION

This is to certify that the work reported in the present project entitled "**IMPLEMENTATION OF VOICE AND GESTURE CONTROL APPLICATION TO SMART ADAPTIVE VEHICLE**" being submitted by **K.V.N.SHARANYA(**1608-19-734-003**), M.ANIL(**1608-19-734-018**), J.AKSHITH KUMAR**(1608-19-734-029) in partial fulfillment for the award of the Degree of Bachelor of Engineering in Electrical and Electronics of Matrusri Engineering college, Hyderabad during 2019-2023, is a record of bonafide work carried out under our guidance and supervision.

The results presented in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

| | |
|---|---|
| K.V.N. SHARANYA | (1608-19-734-003) |
| M. ANIL | (1608-19-734-018) |
| J. AKSHITH KUMAR | (1608-19-734-029) |

Date:

Place: Hyderabad.

## ACKNOWLEDGEMENT

A project is never the outcome of a single person's effort. It is a confluence of varied thought process harmoniously integrated into a resourceful product. It is natural to be indebted to several people for having made this project possible. In the endeavor to do this several people have given their valuable suggestions and guidance at every step. We wish to express our heartful gratitude to each one of them. The accomplishment of this project has been a worthwhile achievement and a learning experience form.

We are thankful to our guide, **Mr**. **GVSSN Srirama Sarma** Assistant Professor, Department of Electrical and Electronics Engineering, MECS for all the guidance and valuable suggestions during the period of our project.

We are extremely thankful to our project coordinators **Dr. P**. **Vasudeva Naidu** Associate professor, Department of Electrical and Electronics Engineering, MECS for their encouragement and support throughout the project.

We would like to express our deepest gratitude to **Prof. G. Ravindranath,** HOD, Department of Electrical and Electronics Engineering, MECS for being a source of inspiration throughout and lending timely help, guidance and support during our project work.

We wish to express our gratitude to **Dr. D. Hanumantha Rao,** Principal Matrusri Engineering College, who permitted us to carry the project work as per the academics.

We are grateful to our supporter **Mr**. **Jeevan,** operation lead in DRDL, for the valuable suggestions and guidance while working on our project.

We are also thankful to all the staff and **Mr. B. Satish Kumar**, lab technician who helped us to complete the project successfully.

Last but not least, friends, well-wishers and other knowledge professionals who have extended their help and advice in making the project successful, we are indebted to all of them. We have a word of appreciation for the co-operation by our parents. We sincerely acknowledge all their efforts which have indispensable to my effort.



K.V.N. SHARANYA (1608-19-734-003)
M. ANIL (1608-19-734-018)
J. AKSHITH KUMAR (1608-19-734-029)

# PROJECT OUTCOMES

**CEE482.1** Solve research problems using appropriate techniques, tools and skills.

**CEE482.2** Design, analyze and evaluate research works.

**CEE482.3** Present project findings effectively and produce technical papers and thesis.

**PROGRAM OUTCOMES (PO's)**

**PO-1(Engineering Knowledge):** Apply knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

**PO-2(Problem Analysis):** Identify, formulate, research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.

**PO-3(Design/ Development of Solutions):** Design solutions for complex engineering problems and design system components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal and environmental considerations.

**PO-4(Conduct investigations of complex problems):** By using the research-based knowledge and research methods including design of experiments, analysis and interpretation of data and synthesis of information to provide valid conclusions.

**PO-5(Modern Tool Usage):** Create, select and apply appropriate techniques, resources and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

**PO-6(The Society):** Apply Engineer and reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice.

**PO-7(Environment and Sustainability):** Understand the impact of professional engineering solutions in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.

**PO-8(Ethics):** Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.

**PO-9(Individual and Team Work):** Function effectively as an individual, and as a member or leader in diverse teams and in multi-disciplinary settings.

**PO-10(Communication):** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations and give and receive clear instructions.

**PO-11(Project Management and Finance):** Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multi-disciplinary environments.

**PO-12(Life-long Learning):** Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

## PROGRAM SPECIFIC OUTCOMES (PSO's)

**PSO-1:** Apply appropriate techniques, hardware, and software tools to design, analyze and test various systems in power systems engineering adaptable to multi-disciplinary environments.

**PSO-2:** Identify the optimal solutions for societal electrical energy requirements by applying suitable design and control strategies.

# TABLE OF CONTENTS

**LIST OF FIGURES**

# LIST OF TABLES

# ABSTRACT

The purpose of robotics in commercial & residential intention has come to be quite essential for executing challenging work in a more conveniently simple way. There is a lot of research work going on to enhance the connection between humans and robots. The main aim of this project is to implement voice and gesture control applications to a smart adaptive vehicle. The existing Bluetooth-controlled vehicle with an embedded system is not suitable for blind or hand-disabled individuals to travel from one place to another. So, this project aims to create a solution for them. The primary objective of this project is to implement voice and gesture commands to control the vehicle. The v3 microphone module is used to record voice commands to send to an Arduino uno microcontroller. For gesture control operation, gyroscope is used to send input commands which are helpful for those who cannot speak. This vehicle can perform various operations viz., forward, backward, left, right, wake-up and shut-down. The existing Bluetooth based controller is upgraded by adding voice and gesture control applications. Sequential operations are performed to switch from one command to another by adding a switching module.

# CHAPTER – 1

# INTRODUCTION

## 1.1 Objectives of present study

The present work is attempting to implement voice and gesture control applications to smart adaptive vehicle.

## 1.1.Organization of thesis

Chapter - 1: Gives the introduction of the projects along with the basis of the problem and solution will be discussed.

Chapter - 2: Deals with the literature survey from different journals and books which contributed in developing the project.

Chapter - 3: Describes the theoretical aspects.

Chapter - 4: Covers the operation of various components.

Chapter - 5: Presents the control of the vehicle using Arduino.

Chapter – 6: Conclusion and references.

## 1.2.Introduction

In today's world, technology has revolutionized the way we live and interact with our surroundings. However, for people with disabilities, certain aspects of daily life can still pose significant challenges. Transportation for the physically disabled can be a major hurdle, as many vehicles are not designed to accommodate their needs. According to the World Health Organization (WHO) and World Bank, there are 70 million disabled people in the world, and a significant proportion of these individuals struggle with mobility issues.

To address this challenge, this project aims to design and implement a smart adaptive vehicle that can be controlled using voice and gesture commands. By utilizing advanced technology, the project seeks to empower disabled individuals to move more freely and independently, without requiring the assistance of others. The proposed vehicle uses an Arduino microcontroller to process voice and gesture commands, and a gyroscope for input commands for those who cannot speak. By integrating these technologies, the vehicle can perform various operations, including moving forward, backward, left, and right, as well as wake-up and shut-down.

The project builds on existing research in the field of robotics and seeks to enhance the connection between humans and robots. The use of voice and gesture commands represents a novel approach to vehicle control, which could have significant implications for disabled individuals worldwide. Through the integration of these technologies, the project aims to improve the quality of life for disabled individuals and promote greater independence and mobility.

Overall, the smart adaptive vehicle project represents an exciting opportunity to leverage advanced technology for social good. By empowering disabled individuals to move more freely and independently, the project has the potential to transform the lives of millions of people around the world.

### 1.3.Problem statement

In the recent world running wholly with technology, physically disabled individuals are facing difficulties with transportation. With the concept of vehicle automation, development of a Bluetooth-controlled vehicle with an embedded system is made to be used by Disabled People not having any Hand Disability to travel from one place to another. The main disadvantage is that it cannot be used by the blind or handicapped persons who cannot use the mobile.

### 1.4.Importance and need of smart vehicle

The importance and advantages of smart vehicle are as follows

- It can be travelled on local roads on ease.
- It is self - operable.
- It does not give any stress to the body.
- All types of disabled people can use this vehicle.
- It is cost effective.

- Mode can be changed based on the requirement of the disabled person.
- It is eco-friendly.

## 1.5.Present Status of Mobility Basis for Disabled People

The present status of mobility aids for disabled people has seen significant advancements and improvements in recent years. There are now a wide variety of mobility aids available to meet the diverse needs of individuals with mobility impairments. some of the latest advancements in mobility aids are Smart wheelchair technology, Prosthetic limbs, Electric mobility scooters, Accessible public transportation.

## 1.6.Summary

This chapter gives a general idea of the project by compelling an introduction to voice and gesture-controlled application to smart adaptive vehicle.

## CHAPTER - 2

## LITERATURE SURVEY

### 2.1 Introduction

A very thorough literature survey is required to know work done earlier in the concerned area. This shall provide the necessary inputs for better understanding of project requirements and to make an expert problem formulation. This is because a research problem description is built on conclusion drawn from the earlier attempts. This chapter presents the papers and journals taken from different publications referred to the project. Conclusions derived from those papers are presented.

### 2.1.Literature

#### 2.2.1 Hand Gesture and Voice Controlled Smart Vehicle

"International Journal of Modern Science and Technology (IJMST)", 2020;5(6):164-167.

ISSN: 2456-023, May 2018.

**Abstract**

This project aimed to develop a robotic vehicle controlled by Arduino using voice-based and hand gesture-based information. The use of Arduino as the central device and the integration of hand gesture technology enabled the vehicle to move wirelessly based on radio frequency signals. Additionally, the project was designed to recognize the commands of the user, making it a voice-controlled vehicle. Overall, the project demonstrated the potential of combining robotics and human-machine interfaces to create efficient and user-friendly vehicles for various commercial and residential applications.

**Conclusion**

The use of voice and gesture control mechanisms in this project, along with the Arduino platform, has made it possible to develop a robotic vehicle that can be controlled with ease. The project demonstrates how Arduino and Radio Frequency (RF) technology can be leveraged to create a hand gesture-controlled vehicle that moves wirelessly. The voice recognition module of the vehicle also allows for voice commands to be recognized, thereby making the vehicle voice controlled. This project highlights the importance of research and development aimed at improving the interaction between humans and robots, as it has significant implications for commercial and residential purposes.

### 2.2.2. Design and Implementation of a Voice Controlled Robot with Human Interaction Ability

"International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering IC4ME2 -2017, 26-27 January 2017"


**Abstract**

The paper presents the research of the designing & development of a voice controlled talking robot using mobile phone based on Arduino Uno microcontroller. The control system of the robot movement will be employed by the voice and the robot will respond to the commanding persons by generating sounds of human voice with each verbal instruction. The proposed system will be designed based on microcontroller which is connected to smart android phone through Bluetooth module for receiving voice command. The voice command is converted to text by an app of the android phone and sends necessary data to the microcontroller for controlling robot movement. After receiving the data, the robot responses according to the command by performing proper movement to the proper direction according to the voice command


**Conclusion**

Voice commands can control the robot to move forward, backward, left and right. There is a voice command "Autonomous" which can instantly make the robot move fully automatically without hitting any obstacle using ultrasonic sensor. Instant stopping of the robot from any kind of movement can be done by the voice command "Stop" at any time. The developed robot can interact with its user using the prerecorded human voice file. For each command, different individual response's audio files are recorded and stored as wav files on SD card. When user commands any instruction the robot will generate the related human voice as response on amplifier from micro-SD card.

### 2.2.3. An Automated Robot-Car Control System with Hand Gestures and Mobile Application Using Arduino

#### Abstract

Gesture recognition has always been a technique to decrease the distance between the physical and the digital world. In this work, we introduce an Arduino based vehicle system which no longer requires manual controlling of the cars. The proposed work is achieved by utilizing the Arduino microcontroller, accelerometer, RF sender/receiver, and Bluetooth. Two main contributions are presented in this work. Firstly, we show that the car can be controlled with hand gestures according to the movement and position of the hand. Secondly, the proposed car system is further extended to be controlled by an android based mobile application having different modes (e.g., touch buttons mode, voice recognition mode). In addition, an automatic obstacle detection system is introduced to improve the safety measurements to avoid any hazards. The proposed systems are designed at lab-scale prototype to experimentally validate the efficiency, accuracy, and affordability of the systems. We remark that the proposed systems can be implemented under real conditions at large-scale in the future that will be useful in automobiles and robotics applications.

#### Conclusion

The proposed work on voice and gesture-based control of a vehicle using Arduino has shown the potential of gesture recognition technology to make vehicle control more effortless and convenient. By utilizing Arduino microcontroller, accelerometer, RF sender/receiver, and Bluetooth, the development of a system has been made that allows users to control a car with hand-gestures and through an android-based mobile application. Through this project, we have learned about the efficiency, accuracy, and affordability of the proposed systems, and their potential applications in the automobile and robotics industries. Overall, this project has contributed to our understanding of the possibilities of gesture recognition technology in developing intelligent vehicle systems.

### 2.3. Summary

After analyzing all these technical papers, the proposed model of voice and gesture-controlled application to a smart adaptive vehicle using Arduino can be developed.

# CHAPTER - 3

# THEORETICAL ANALYSIS

## 3.1 Introduction

This chapter covers all the project's technical concepts regarding the tools and methods used. It also includes the fundamental information on how the project's components were selected. The chapter will provide the reader with useful background knowledge on the project before they move on to the experimental analysis.

## 3.2 Power supply

A power supply is an electrical device that provides electric power to an electrical load. Its primary function is to convert electric current from a source to the correct voltage, current, and frequency required to power the load. Power supplies are also referred to as electric power converters as they convert one form of electrical energy into another. In general, power supplies can be standalone pieces of equipment or built into the load appliances they power. For example, power supplies in desktop computers and consumer electronics are built into the devices they power.

Power supplies perform a variety of functions, including limiting the current drawn by the load to safe levels, shutting off the current in the event of an electrical fault, and power conditioning to prevent electronic noise or voltage surges from reaching the load. Power-factor correction and energy storage are also functions performed by some power supplies, which allow them to continue powering the load in the event of a temporary interruption in the source power. All power supplies have a power input connection that receives energy in the form of electric current from a source and one or more power output connections that deliver current to the load. The source power may come from the electric power grid, such as an electrical outlet, energy storage devices such as batteries or fuel cells, generators or alternators, solar power converters, or another power supply. The input and output connections are usually hardwired circuit connections, although some power supplies use wireless energy transfer to power their loads without wired connections. Some power supplies have additional inputs and outputs for external monitoring and control. Here, we are using batteries as the power supply for our project.

### 3.2.1 Batteries

A battery is an electrochemical device that stores chemical energy and converts it into electrical energy. It consists of one or more electrochemical cells, which are composed of two electrodes, a cathode, and an anode, separated by an electrolyte. When a battery is connected to an external load, a chemical reaction occurs in the cell, causing the flow of electrons from the anode to the cathode through the external circuit, producing an electrical current.

Batteries are commonly used as power sources for a wide range of devices, including flashlights, mobile phones, electric cars, and more. They can be classified into two types: primary batteries and secondary batteries. Primary batteries, also known as disposable batteries, are designed to be used once and then discarded, as their electrodes are irreversibly changed during discharge. Common examples of primary batteries include alkaline batteries used in flashlights and portable electronic devices. Secondary batteries, also known as rechargeable batteries, can be recharged, and used multiple times by applying an electric current to reverse the chemical reaction. Examples of secondary batteries include lead-acid batteries used in vehicles and lithium-ion batteries used for portable electronics such as laptops and mobile phones.

Batteries come in a variety of shapes and sizes, from small cells used in hearing aids and wristwatches to large batteries used in vehicles and backup power systems. They have lower specific energy (energy per unit mass) than common fuels such as gasoline. However, in electric vehicles, the higher efficiency of electric motors in converting electrical energy to mechanical work makes up for this drawback. Taking cost into consideration, it is decided to use lead-acid batteries as they are more compatible and desirable for this project.

Lead acid battery of 12V operating voltage and 42 Ah current per hour as shown in figure 3.1 is used in this project.


**Figure 3.1 Battery**

### 3.3 Arduino UNO

The Arduino Uno shown in the figure 3.2 is an open-source microcontroller board based on the MicrochipATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is like the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. LA Wet and production files for some versions of the hardware are also available. The word "uno" means "one" in Italian and was chosen to mark the initial release of Arduino Software. The Uno board is the first in a series of USB-based Arduino boards; it and version 1.0 of the Arduino IDE were the reference versions of Arduino, which have now evolved to newer releases. The ATmega328 on the board comes preprogrammed with a boot loader that allows uploading new code to it without the use of an external hardware programmer.

While the Uno communicates using the original STK500 protocol, it differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. The Arduino Uno is depicted as figure 3.2.

**Figure 3.2 Arduino Uno**

Arduino can be used to communicate with a computer, another Arduino board, or other microcontrollers. The ATmega328P microcontroller provides UART TTL (5V) serial communication which can be done using digital pin 0 (Rx) and digital pin 1 (Tx). An ATmega16U2 on board channels this serial communication over USB and appears as a virtual com port to software on the computer. The ATmega16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. There are two RX and TX LEDs on the Arduino board which will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (not for serial communication on pins 0 and 1). A Software Serial library allows for serial communication on any of Uno's digital pins. The ATmega328P also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus.

### 3.3.1 Features of Arduino Uno

- The main features of Arduino UNO are listed below.
- Microcontroller: ATmega328
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 40 mA

- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader.
- SRAM: 2 KB (ATmega328)

### 3.3.2 Applications of Arduino Uno

The applications of Arduino UNO are as follows.

- Embedded Systems.
- Automation.
- Robotics.
- Control Systems.
- Instrumentation.
- Prototyping of Electronics Products and Systems
- Multiple DIY Projects.
- Easy to use for beginner level DIYers and makers.
- Projects requiring Multiple I/O interfaces and communications.

### 3.3.1 Arduino UNO pin diagram

The pin diagram of Arduino UNO is shown in the figure 3.3 below

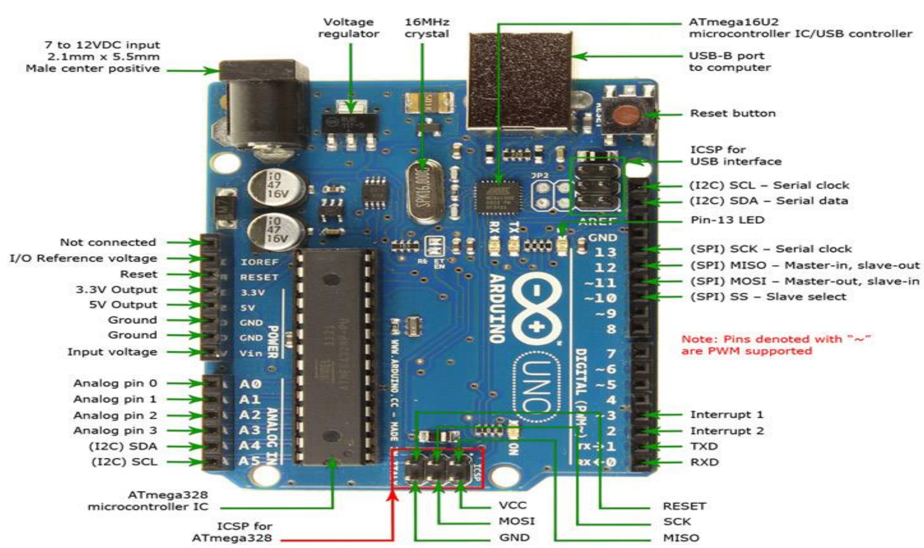


**Figure 3.3 Arduino Uno pin Diagram**

### 3.3.4 Pin Description

The description of each pin in Arduino UNO is discussed in tabular column Table 3.1

| **Pin Category** | **Pin Name** | **Details** |
|---|---|---|
| Power | Vin, 3.3V, 5V, GND | Vin: Input voltage to Arduino when using an external power source.<br><br>5V: Regulated power supply used to power microcontroller and other components on the board.<br><br>3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA.<br><br>GND: ground pins. |
| Reset | Reset | Resets the microcontroller. |
| Analog Pins | A0 – A5 | Used to provide analog input in the range of 0-5V |
| Input / Output Pins | Digital Pins 0 – 13 | Can be used as input or output pins. |
| Serial | 0(Rx), 1(Tx) | Used to receive and transmit TTL serial data. |
| External Interrupts | 2, 3 | To trigger an interrupt. |
| PWM | 3, 5, 6, 9, 11 | Provides 8-bit PWM output. |
| SPI | 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK) | Used for SPI communication. |
| Inbuilt LED | 13 | To turn on the inbuilt LED. |
| TWI | A4 (SDA), A5 (SCA) | Used for TWI communication. |
| AREF | AREF | To provide reference voltage for input voltage. |

**Table 3.1 Pin Description of Arduino**

### 3.3.5 Arduino IDE

- *Auto Format*: This formats your code nicely: i.e., indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more.

- *Archive Sketch* Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.

- *Fix Encoding & Reload* Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.

- *Serial Monitor* Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.

- *Port* This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.

- *Programmer* For selecting a hardware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a bootloader to a new microcontroller. *The arduino IDE is depicted in figure 3.4*

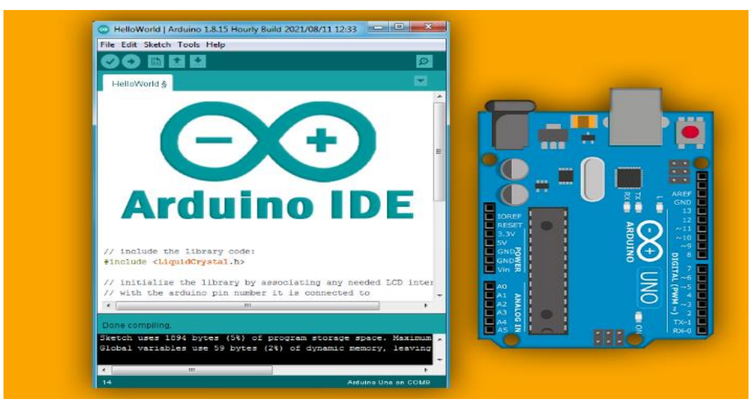


**Figure 3.4 Arduino IDE**

### 3.4 Driver Circuit

The driver circuit is a crucial component in controlling the operation of various electrical devices, such as motors, displays, and stepper motors. In the current project, the driver circuit is responsible for controlling the direction of two DC motors simultaneously, thereby ensuring synchronous movement of the vehicle wheels. This is achieved by utilizing two MOSFETs and two diodes in the circuit.

The main function of a driver circuit is to control the behaviour of another circuit or component, such as regulating the current flowing through a circuit or controlling various factors that affect the performance of other components. This can be accomplished using specialized integrated circuits, which are designed to control high-power switches in switched-mode power converters, for example. In addition, amplifiers can also be considered drivers for loudspeakers, while voltage regulators can serve as drivers for components operating within a broad range of input voltages.

#### 3.4.1 PWM Motor Driver

PWM (Pulse Width Modulation) motor drivers are electronic devices used to control the speed and direction of DC motors using a PWM signal. PWM is a technique where the duty cycle of a rectangular wave is modulated to control the average voltage applied to the motor. The duty cycle represents the fraction of time during which the signal is high, which in turn determines the average voltage applied to the motor.

PWM motor drivers can be implemented using various types of electronic components such as transistors, MOSFETs, or H-bridge ICs. The basic function of a PWM motor driver is to convert a low-power control signal (such as from a microcontroller) into a high-power signal that can drive the motor.

PWM motor drivers are commonly used in robotics, automation, and other applications where precise control of motor speed and direction is required. They offer several advantages over traditional linear motor drivers, including better efficiency and lower heat dissipation.

When choosing a PWM motor driver, some important factors to consider include the maximum current and voltage ratings, the PWM frequency range, and thermal characteristics. Some popular PWM motor driver ICs include the L293D, L298N, and DRV8833, among others.

### 3.4.2. Specifications of PWM Motor Driver

**Input Voltage Range**: The range of input voltage that the motor driver can accept. It typically varies from a few volts to several tens of volts.

**Output Current:** The maximum continuous current that the motor driver can deliver to the motor. It is usually specified in amps (A) or milliamps (mA).

**PWM Frequency**: The frequency at which the motor driver generates the pulse-width-modulated signals. It is often measured in kilohertz (kHz) or megahertz (MHz).

**Control Interface**: The type of control interface provided by the motor driver to adjust the motor speed and direction. It can be analog, digital, or a combination of both. Common control interfaces include analog voltage, serial communication (UART, I2C, SPI), or PWM input.

**Motor Type:** The types of motors that the driver is designed to operate. It can be DC motors, stepper motors, or brushless DC motors. The motor driver's specifications should match the requirements of the motor you intend to control.

**Protection Features**: Motor drivers often include protection features to prevent damage to the motor and the driver itself. These features can include overcurrent protection, over-temperature protection, short-circuit protection, and reverse voltage protection.

**Efficiency:** The efficiency of the motor driver represents how effectively it converts the input power into motor output power. It is usually given as a percentage.

**Size and Form Factor**: The physical dimensions and form factor of the motor driver. This information is essential to ensure compatibility and integration with your application.

**Additional Features**: Some motor drivers may offer additional features such as fault detection, braking capabilities, soft-start functionality, and programmability. The PWM motor is shown in figure 3.5.

**Figure 3.5 PWM Motor Driver**

### 3.4.3 Specifications and Ratings of PWM Motor

The ratings of PWM motor driver is described in below table 3.2

| **Specification** | **Ratings** |
|---|---|
| Input voltage range | 6V-24V |
| Output Current | Upto 2A |
| PWM Frequency | 10Khz-100Khz |
| Control Interface | PWM, UART |
| Efficiency | 90% |
| Protection Features | Over current, short circuit, Over temperature |
| Size and Form factor | 25 mm* 25 mm |

**Table 3.2 Specifications of PWM Motor Driver**

### 3.5 Buck Converter

A buck converter (step-down converter) is a DC-to-DC power converter which steps down voltage (while drawing less average current) from its input (supply) to its output (load). It is a class of switched-mode power supply (SMPS) typically containing at least two semiconductors (a diode and a transistor, although modern buck converters frequently replace the diode with a second transistor used for synchronous rectification) and at least one energy storage element, a capacitor, inductor, or the two in combination. To reduce voltage ripple, filters made of capacitors (sometimes in combination with inductors) are normally added to such a converter's output (load-side filter) and input (supply-side filter). It is called a buck converter because the voltage across the inductor “bucks” or opposes the supply voltage. The buck converter is depicted in figure 3.6.


**Figure 3.6 Buck converter**

#### 3.5.1 Specifications of DC-DC Buck Converter

Conversion efficiency: 92%(highest)

Switching frequency: 150KHz.

Output ripple: 30mA9maximum)

Load Regulation: ± 0.5%

Voltage Regulation: ± 0.5%

Dynamic Response speed: 5% 200uS.

Input voltage:4.75-35V.

Output voltage:1.25-26V(Adjustable)

### 3.5.2. Advantages of Buck Converter

**High Efficiency:** Buck converters are known for their high efficiency. They can efficiently step down the input voltage to a lower output voltage without significant power loss. This efficiency is achieved using a power switch (typically a transistor) and an inductor, which helps regulate the output voltage.

**Voltage Regulation**: Buck converters provide excellent voltage regulation. They maintain a stable output voltage even when the input voltage or load conditions vary. This stability is crucial in many electronic systems to ensure proper operation of sensitive components.

**Compact size:** Buck converters are typically compact in size due to their simplified circuitry. They require fewer components compared to other types of converters, making them suitable for applications where space is limited.

**Low Cost**: The simplicity of the buck converter's design leads to a lower manufacturing cost compared to other types of converters. This makes them an economical choice for many applications.

**Fast Response:** Buck converters can respond quickly to changes in load conditions. They have a fast transient response, meaning they can adjust the output voltage rapidly when the load changes. This characteristic is essential in applications where rapid voltage regulation is required.

**Wide Range of Applications**: Buck converters are versatile and find applications in various fields, including power supplies for computers, consumer electronics, automotive systems, LED lighting, battery chargers, and more. Their ability to efficiently step-down voltage while maintaining voltage regulation makes them suitable for a wide range of electronic devices.

**Reduced Heat Dissipation:** Due to their high efficiency, buck converters generate less heat compared to other types of converters. This not only contributes to their overall efficiency but also helps reduce the size and cost of heat sinks or cooling mechanisms in the system.

**Energy Conservation**: Buck converters can contribute to energy conservation by efficiently converting power from a higher voltage source to a lower voltage, reducing energy waste in the process. This efficiency is especially beneficial in battery-powered systems where maximizing battery life is crucial.

### 3.6 Microphone V3 Module

The Microphone V3 module is an electronic component that converts sound waves into electrical signals. It is widely used in various applications such as audio recording, speech recognition, and noise cancellation.

The module has a small PCB (Printed Circuit Board) with a built-in electret condenser microphone and a preamplifier circuit. The electret condenser microphone has a built-in capacitor that provides a stable bias voltage for the microphone's internal amplifier. The preamplifier circuit amplifies the microphone signal and converts it into an analog output signal that can be easily read by a microcontroller or other digital devices.

The Microphone V3 module operates on a voltage range of 3.3V to 5V DC and has a frequency response range of 20Hz to 20kHz. The module has a small form factor, making it suitable for use in compact electronic projects.The module is easy to use and can be connected directly to a microcontroller or other digital devices using its three-pin interface. The pinout includes a ground pin, a power supply pin, and an analog output pin. The analog output pin can be connected to an analog-to-digital converter (ADC) input of a microcontroller to read the sound signal.

The microphone v3 module is depicted in figure 3.7.

**Figure 3.7 Microphone V3 Module**

### 3.6.1 Specifications of V3 Module

Voltage: 4.5-5.5V

Current: <40mA

## 3.7 ADXL335 Accelerometer

The ADXL335 is a highly regarded 3-axis accelerometer manufactured by Analog Devices. It is widely used in numerous applications, ranging from consumer electronics to industrial systems. The accelerometer offers the ability to measure both static and dynamic acceleration in three axes: X, Y, and Z. With its selectable measurement range of ±3g, ±6g, or ±9g, the ADXL335 can capture accelerations up to three times the acceleration due to gravity in each direction. Its sensitivity is adjustable based on the chosen range, typically around 330 mV/g.

This accelerometer provides analog voltage outputs that are proportional to the acceleration along each axis. The ADXL335 operates on low power, consuming less than 350 microamps during active measurement. It boasts a wide temperature range of -40°C to +85°C, allowing for deployment in various environmental conditions. Additionally, the accelerometer incorporates a self-test feature to verify its functionality, and its low noise design ensures accurate and reliable acceleration measurements.

Featuring a compact surface-mount package, the ADXL335 is well-suited for integration into space-constrained designs. It finds applications in fields such as robotics, gaming, fitness tracking, motion detection, and gesture recognition. Analog Devices provides comprehensive documentation, including datasheets and application notes, along with development tools and evaluation boards to aid users in effectively utilizing the ADXL335 accelerometer. Calibration may be necessary for precise measurements, and following the manufacturer's guidelines is essential for optimal implementation of this versatile sensor. The below figure 3.8 depicts the ADXL335 Accelerometer.

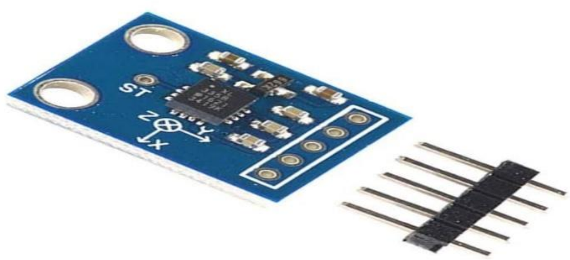

**Figure 3.8 ADXL335 Accelerometer**

### 3.7.1 Specifications of Accelerometer

Measurement range: ±3 g (where g is the acceleration due to gravity, approximately 9.81 m/s^2)

Sensitivity: 330 mV/g

Supply voltage: 1.8 V to 3.6 V

Current consumption: 350 µA (typical)

Output voltage range: 0.3 V to (Vcc - 0.3 V)

Bandwidth: 0.5 Hz to 1600 Hz (-3 dB)

### 3.7.2 Advantages of Accelerometer

**Bandwidth**: The ADXL335 has a bandwidth range of 0.5 Hz to 1 kHz, which means it can effectively capture acceleration changes within that frequency range.

**Temperature Range:** The accelerometer can operate within a wide temperature range, typically from -40°C to +85°C, making it suitable for both indoor and outdoor applications.

**Self-Test Feature:** The ADXL335 includes a self-test feature that allows for verification of the sensor's functionality. When enabled, it applies a known electrostatic force to each axis and allows you to check if the accelerometer responds accurately.

**Low Noise:** The ADXL335 is designed with low noise characteristics, which helps to improve the accuracy and reliability of the measured acceleration values.

**Robust Construction:** The accelerometer is built with durable materials and features robust construction, making it resistant to mechanical stress, shocks, and vibrations.

**Supply Voltage:** The ADXL335 operates with a supply voltage range of 1.8V to 3.6V, providing flexibility for various power supply configurations.

**Support and Documentation:** Analog Devices provides comprehensive documentation, including datasheets, application notes, and reference designs, to assist users in implementing the ADXL335 in their projects.

**Development Tools:** Analog Devices also offers development tools and evaluation boards specifically designed for the ADXL335, facilitating rapid prototyping and evaluation of the accelerometer's performance.

**Calibration:** Like any accelerometer, the ADXL335 may require calibration to ensure accurate and precise measurements. Calibration processes typically involve applying known forces or accelerations to the device and adjusting the output accordingly.

### 3.8 Relay Module

A 4-channel relay module is an electronic module that can control up to four individual electrical circuits using a single microcontroller or other electronic device. Each channel typically has a relay that can switch an electrical load on or off, such as a motor, light, or other device. The module is often used in home automation systems, robotics projects, and other electronic projects that require the ability to control multiple electrical circuits. Here are some key features and advantages of a 4-channel relay module:

**Control Multiple Circuits:** With four independent relays, the module enables simultaneous control of up to four different electrical circuits. This flexibility is useful in various applications where multiple devices or systems need to be switched on or off simultaneously or independently.

**Easy Interface:** The module is designed to interface easily with microcontrollers, Arduino boards, or other control systems. It typically uses digital input signals to trigger the relays, making it straightforward to integrate into existing projects or systems.

**Versatile Voltage Compatibility:** 4-channel relay modules are available in different voltage variants, such as 5V or 12V, allowing compatibility with various voltage requirements. This flexibility enables the control of different types of devices or systems with varying voltage levels.

**High Switching Capacity:** Relay modules often have a high switching capacity, allowing them to handle substantial electrical loads. This feature makes them suitable for controlling devices such as motors, lights, solenoids, or other high-power components.

**Opto-isolation**: Many 4-channel relay modules feature opto-isolation for enhanced protection. Opto-isolation provides electrical isolation between the control signal and the relay's switching circuit, helping to prevent voltage spikes or electrical noise from affecting the control system.

**Status Indication:** Some relay modules include LED indicators for each relay channel. These LEDs provide visual feedback on the status of each relay, indicating whether it is active or inactive. This feature simplifies troubleshooting and monitoring of the controlled circuits.

**Compact and Convenient:** The relay modules are usually compact in size, allowing for space-efficient installation. They often come with screw terminals or header pins, facilitating easy wiring and connection of external devices.

**Wide Range of Applications**: 4-channel relay modules find applications in home automation, industrial automation, robotics, IoT projects, smart appliances, and more. They are commonly used in situations that require centralized control of multiple electrical circuits. The relay module is depicted in figure 3.9 below.

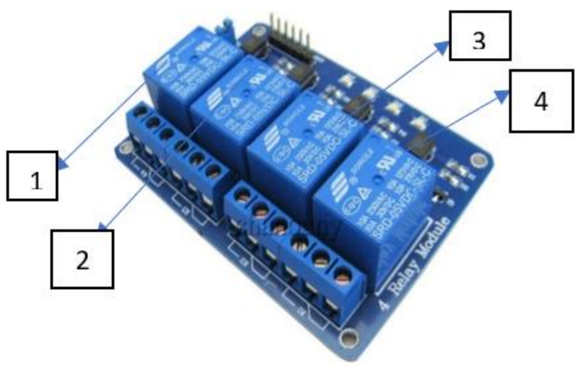


**Figure 3.9 Relay module**

The Relay module represents:

1. Forward direction relay
2. Backward direction relay
3. Left Relay module.
4. Right Relay module

### 3.8.1 Specifications of Relay Module

Voltage rating: 5V DC

Maximum switching voltage: 250V AC / 30V DC

Maximum switching current: 10A

Trigger current: 15-20mA

Relay type: electromagnetic

Number of channels: 4

Input signal: active low (LOW triggers the relay)

Header pins for easy connection to microcontrollers or other devices

LED indicators for each channel to show the status of the relay (on/off)

### 3.9 DC Motors

DC motors are devices that convert electrical signals into mechanical energy. Specifically, a Permanent Magnet DC Motor (PMDC) is a type of DC motor that incorporates a permanent magnet pole. In this motor, the magnet is used to generate the flux in the air gap, rather than using a field winding. The rotor structure of a PMDC motor is like that of a conventional DC motor and includes an armature core, commutator, and armature winding. The motor has two types of windings: the field winding and the armature winding. The field winding's primary function is to produce the working magnetic flux in the air gap and is wound on the stator of the motor. The armature winding, on the other hand, is wound on the rotor.

#### 3.9.1 Calculations for motor selection

- We have done a small survey for deciding the most preferable vehicle speed (V). The result obtained was 15 kmph.
- The constant used for converting linear velocity to angular velocity is 9.5493.
- The wheel radius ($R_w$) which is considered approximately 20 cm i.e., 0.2 m.
- By using the below formula, we can calculate the motor speed required for the vehicle.

The figure 3.10 depicts the DC motor.

**Figure 3.10 DC Motor**

$$Motorspeed(rpm) = \frac{V * 9.5493}{3.6 * Rw}$$

$$= \frac{15 * 9.5493}{3.6 * 0.2} \sim 200 \text{ rpm}$$

**Table 3.3 motor readings**

| **Parameters** | **Readings given** |
|---|---|
| Operating Voltage Range | 6V to 15V DC |
| Recommended Voltage (DC) | 12V |
| Rated RPM (at 12V) | 200 |
| Rated Torque (Kg-cm) | 4 |
| Full (Stall) Load Torque (Kg-cm) | 20 |
| No- Load Current (A) | 0.2 |
| Full Load Current (A) | 1.2 |
| Base Motor Dimensions (mm) | 35.5 x 58 (Diameter x Length) |
| Gearbox Dimensions (mm) | 50.5 x 34.5 (Diameter x Length) |
| Motor Shaft Length (mm) | 21 |
| Shaft Diameter (mm) | 8 |
| Motor Shaft Type | D |
| Weight (gm) | 292 |

Taking the motor speed into consideration, the motor in figure 3.10 is procured and the motor readings are obtained as in table 3.3.

Power = V * I = 12 * 0.2 = 2.4 Watts (No load)

Power = V * I = 12 * 0.2 = 14.4 Watts (Full load)

### 3.9.2 Specifications of DC Motor

Power: 250 W

Voltage: 24V

Speed: 700-1000 rpm

Rated torque:1.0-13.0 in-lbs.

## 3.10 Summary

This chapter has dealt with all about the basic hardware required for developing prototype model and the basis of choosing each component of the adaptive automobile. The following chapter will deal with the setup of these components for testing and evaluation.

# CHAPTER - 4

# EXPERIMENTAL ANALYSIS

## 4.1 Introduction

The theory in the previous chapter is entirely related to the practical setup of the project, which is detailed in this chapter. This chapter also includes the kit's experimental tests.

### 4.2 Methodology

The project aims to develop a smart adaptive vehicle that can be controlled through voice and gesture commands, making it more accessible for blind or hand-disabled individuals to travel from one place to another. The vehicle is based on an existing Bluetooth-controlled platform which is upgraded by adding a microphone module for voice commands and a gyroscope module for gesture control.

The V3 module is used to record voice commands, which are then sent to an Arduino Uno microcontroller via Arduino IDE. The gyroscope module captures the user's hand movements, which are processed and sent to the Arduino Uno. The signal is then received by the motor driver to rotate the motors, resulting in the desired movement of the vehicle.

Sequential operations are performed by adding a switching module, allowing the user to switch between different commands seamlessly.

The project's methodology involves a combination of hardware components and software programming, which allows for intuitive and accessible control of the vehicle. The microphone and gyroscope modules capture user inputs and convert them into signals that are processed by the microcontrollers and transmitted to the motor driver, resulting in the vehicle's movement.

## 4.3 Block Diagram

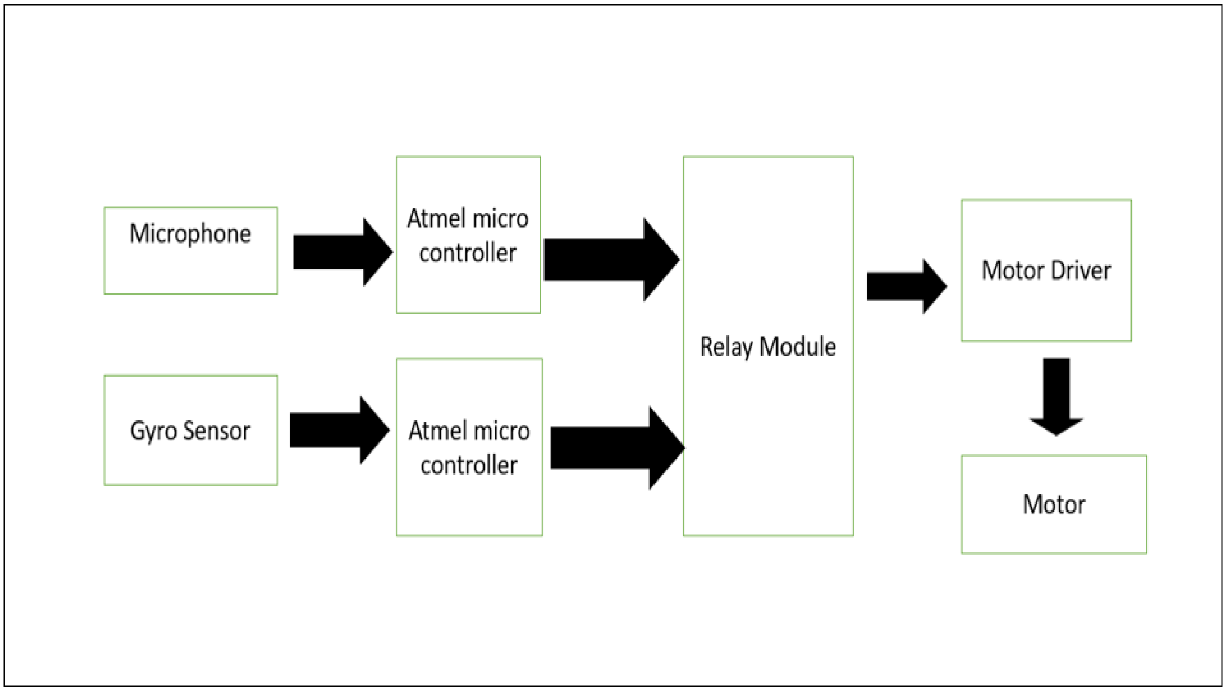


**Figure 4.1 Block diagram**

Figure 4.1 depicts the comprehensive block diagram showcasing the operation of voice and gesture control applications. Initially, the inputs received from a microphone or gyro sensor are transmitted to an Arduino Uno using the Arduino IDE. These input signals, encoded as code, are then forwarded to a Relay module, responsible for sensing the input. Subsequently, the processed inputs are directed to a motor driver, initiating the operation of the vehicle.

## 4.4 Control Circuit Setup

The complete interfaced circuit diagram i.e., Arduino microcontroller interfaced to Microphone and gyro sensor module and motor driver circuit is shown in figure 4.2.

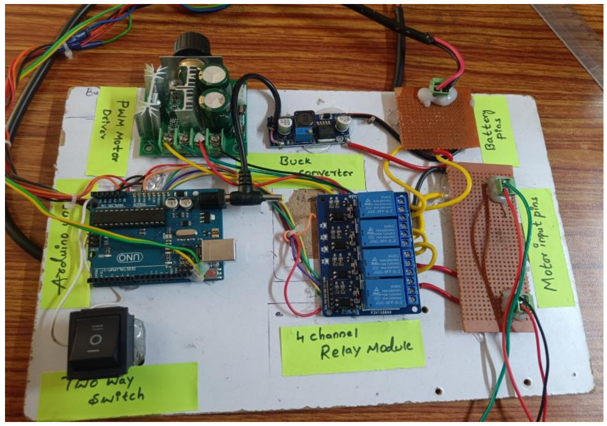


**Figure 4.2 Control Circuit Diagram**

## 4.5 Arduino functioning

The 14 digital input/output pins can be used as input or output pins by using pin Mode (), digital Read () and digital Write () functions in Arduino programming. Each pin operates at 5V and can provide or receive a maximum of 40mA current and has an internal pull-up resistor of 20-50 kohms which are disconnected by default. Out of these 14 pins, some pins have specific functions as listed below:

- **Serial Pins 0 (Rx) and 1 (Tx):** Rx and Tx pins are used to receive and transmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.
- **External Interrupt Pins 2 and 3:** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **PWM Pins 3, 5, 6, 9 and 11:** These pins provide an 8-bit PWM output by using analog Write () function.
- **SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK):** These pins are used for SPI communication.
- **In-built LED Pin 13:** This pin relates to a built-in LED, when pin 13 is HIGH – LED is on and when pin 13 is LOW, its off.

Along with 14 Digital pins, there are 6 analog input pins, each of which provide 10 bits of resolution, i.e., 1024 different values. They measure from 0 to 5 volts, but this limit can be increased by using AREF pin with analog Reference () function.  Analog pin 4 (SDA) and pin 5 (SCA) also used for TWI communication using Wire library.

Arduino Uno has a couple of other pins as explained below:

- **AREF:** Used to provide reference voltage for analog inputs with analogReference() function.
- **Reset Pin:** Making this pin LOW, resets the microcontroller.

When ATmega328 chip is used in place of Arduino Uno, or vice versa, the image below Figure 4.3 shows the pin mapping between the two.

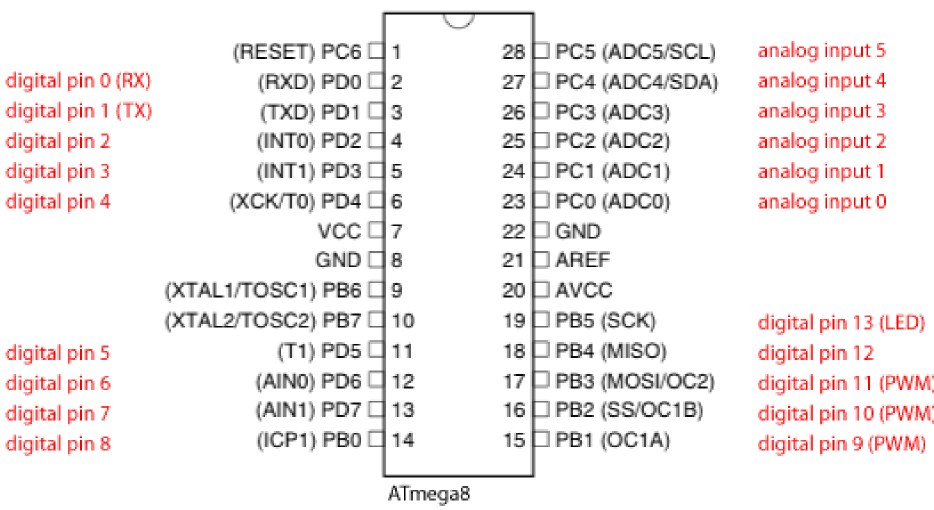


**Figure 4.3 Pin mapping**

### 4.6 Motor Driver Functioning

A PWM system consists of a control circuit, a comparator, a filter, a driver circuit, and a load. By varying the duty cycle of the periodic waveform, the PWM system can regulate the power delivered to the load, providing precise control over its speed or brightness. it is responsible for controlling the speed and direction of the motor that powers the vehicle's movement. The PWM motor driver receives signals from the microcontroller, which is connected to the v3 microphone module and the gyroscope. These signals are processed by the microcontroller to determine the appropriate duty cycle for the pulses that are sent to the motor driver.

When the user gives a voice command or a gesture to control the vehicle's movement, the microcontroller sends a signal to the motor driver to adjust the voltage and current supplied to the motor. The PWM motor driver then uses pulse width modulation to regulate the amount of power that is delivered to the motor. The duty cycle of the pulses determines the average voltage and current that is supplied to the motor, and thus the speed of the vehicle. By reversing the polarity of the voltage that is supplied to the motor, the motor driver can also change the direction of the motor's rotation. This feature is essential for controlling the movement of the vehicle in different directions, such as forward, backward, left, and right.

The PWM motor driver is a crucial component, as it enables precise control over the speed and direction of the motor that powers the vehicle's movement. By incorporating voice and gesture control applications, this smart adaptive vehicle provides a convenient and accessible mode of transportation for blind or hand-disabled individuals. The PWM motor driver allows the user to control the vehicle's movement with ease, making it an integral part of this innovative project. The internal connections of PWM motor are shown in the below figure 4.4.

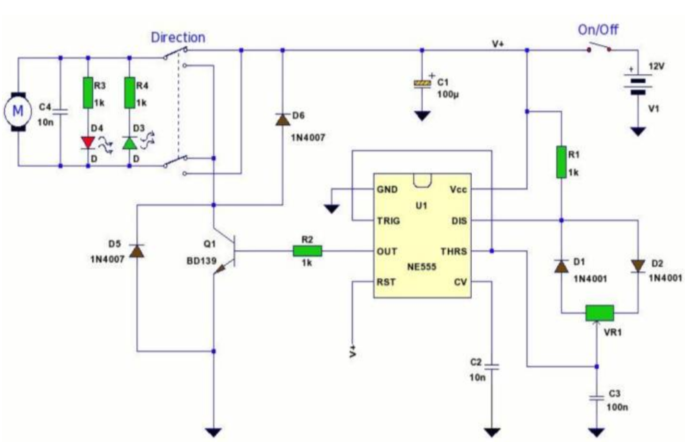


**Figure 4.4 Internal Connections of PWM Motor**

### 4.7 Sequential Switching Operation

Sequential operations are performed to switch from one command to another by adding a switching module. These switching operations are performed by changing the mode of usage. The switching is done between existing Bluetooth-controlled module, voice, and gesture control modules.

#### 4.7.1 Existing Bluetooth based Automobile Working

This project describes the development of a motorized and Bluetooth-controlled vehicle running with an embedded system. The controller used to control an adaptive automobile is an android device which has customized Bluetooth feature built in to communicate with Arduino through Bluetooth module. The task of controlling the vehicle is done by the Arduino UNO which houses the micro- controller ATMEGA32. Two motors are used in the project, one for direction whereas the other for the movement of the vehicle. The motor rotates based on the command resulting in the movement of the vehicle in the desired direction and motion. The developed project is used for elderly and handy capped people who are facing difficulties to travel on their own for small distances. This vehicle has a variety of application areas like hospitality activity, airports, railway platforms, Institutional groups, and several other firms. The control circuit for Bluetooth module is depicted in figure 4.5 and the prototype of Bluetooth module working is shown in figure 4.6.


**Figure 4.5 Bluetooth module functioning**

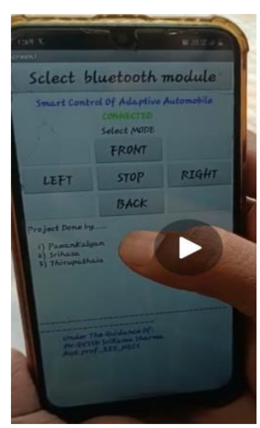


**figure 4.6 Bluetooth working**

1. Arduino UNO
2. L298N motor driver
3. HC – 05 Bluetooth module
4. Relays
5. Bread board
6. Jumper wires

The project is controlled using a smart phone by the user with ease simply by sitting on the chair arranged on the vehicle. The signal given through the mobile phone to the Bluetooth module is sent to the Arduino UNO. Arduino UNO finds the command given through the signal with the help of code dumped and gives it to the driver which is responsible for the rotation of the motor. The motor rotates based on the command resulting in the movement of the vehicle in the desired direction and motion.

### 4.7.2 Voice Controlled Module Functioning

One of the most portable and user-friendly voice recognition modules in the market is Elec House V3. This module can be used in one of two ways: either through the built-in GPIO pins or the serial port. Up to 80 voice commands with a duration of 1500 milliseconds can be stored on the V3 board. Instead of converting user orders to text, this one will compare them to a library of voices that have already been recorded. Therefore, this product can be used without any linguistic limitations. Training is necessary before allowing it to recognize any spoken commands.

The device works at an input voltage range of 4.5 - 5 volts and will draw a current of less than 40 mA. This module can work with 99% recognition accuracy if it is used under ideal conditions. The choice of microphone and the noise in the environment plays a vital role in affecting the performance of the module. Considering this factor, we have chosen this microphone as it has good sensitivity, and it reduces the noise in background while giving commands to get the maximum performance out of the module.

The connections are made as described below.

- GND - Ground
- VCC - 5 V
- RXD - Digital pin 3 of Arduino (This is a user defined pin. Sample code has Pin 3 as Tx.)
- TXD - Digital pin 2 of Arduino (This is also a user defined pin.)

The LED is connected to the digital pin 13 of the Arduino as defined in the sample code. Connect a 470 ohms resistor in series to the LED. Plug the microphone in to the 3.5 mm jack in the board. Solder it to the mic pins in the module if it doesn't come with a 3.5 mm plug.

**4.7.3 Various operations of vehicle**

Each kind of possible operation of the motors harmoniously, results in different directions of the vehicle movement. The operations along with the code and working of the motor is given below.

The forward operation is performed such as

```
void MotorForward()
{
  digitalWrite(RightMotorForward, HIGH);
  digitalWrite(RightMotorBackward, LOW);
  digitalWrite(LeftMotorForward, HIGH);
   digitalWrite(LeftMotorBackward, LOW);}
```

The function sets the digital output of four pins connected to the motor driver module, which controls the direction of the motors. The pins are named RightMotorForward, RightMotorBackward, LeftMotorForward, and LeftMotorBackward.

The code sets the RightMotorForward and LeftMotorForward pins to HIGH, which sets the motor driver module to move the motors in the forward direction. The RightMotorBackward and LeftMotorBackward pins are set to LOW, which disables the backward motion of the motors. This results in both motors moving forward at the same speed, causing the robot or vehicle to move forward.

The Backward operation is as follows:

```
void MotorBackward() {

    digitalWrite(RightMotorForward, LOW);

  digitalWrite(RightMotorBackward, HIGH);

  digitalWrite(LeftMotorForward, LOW);

   digitalWrite(LeftMotorBackward, HIGH);

}
```

This function is similar to the MotorForward() function, but the directions of the motor pins are reversed. In this function, the digital pin for the right motor forward is set to LOW using digitalWrite(RightMotorForward, LOW), which makes the motor rotate in the opposite direction. The digital pin for the right motor backward is set to HIGH using digitalWrite(RightMotorBackward, HIGH), which makes the motor move backward. Similarly, the digital pin for the left motor forward is set to LOW using digitalWrite(LeftMotorForward, LOW), and the digital pin for the left motor backward is set to HIGH using digitalWrite(LeftMotorBackward, HIGH).

So, when this function is called, the motors connected to the pins RightMotorForward, RightMotorBackward, LeftMotorForward, and LeftMotorBackward will rotate in the backward direction.

The code for left movement is as follows:

```
void TurnLeft() {

  digitalWrite(RightMotorForward, HIGH);

  digitalWrite(RightMotorBackward, LOW);

  digitalWrite(LeftMotorForward, LOW);

   digitalWrite(LeftMotorBackward, HIGH);

}
```

This function is likely used in a program that controls a robot or a vehicle with two wheels or motors, where turning left is required. The function sets the pin values of the four motor control pins to specific values. digitalWrite() is an Arduino function that writes a digital value (either HIGH or LOW) to a specified pin.In this function, digitalWrite(RightMotorForward, HIGH) sets the pin connected to the forward motion of the right motor to HIGH, which makes the motor turn forward. Similarly, digitalWrite(LeftMotorBackward, HIGH) sets the pin connected to the backward motion of the left motor to HIGH, which makes the motor turn backward.

At the same time, digitalWrite(RightMotorBackward, LOW) sets the pin connected to the backward motion of the right motor to LOW, which stops the backward motion of the right motor. Likewise, digitalWrite(LeftMotorForward, LOW) sets the pin connected to the forward motion of the left motor to LOW, which stops the forward motion of the left motor. This combination of motor control pin values causes the vehicle to turn left.

The code for right movement is as below:

```
void TurnRight() {
    digitalWrite(RightMotorForward, LOW);
  digitalWrite(RightMotorBackward, HIGH);
  digitalWrite(LeftMotorForward, HIGH);
   digitalWrite(LeftMotorBackward, LOW);
}
```

The function achieves this by controlling the direction of rotation of the two motors connected to the robot. The function code sets the RightMotorForward pin to LOW, which means the right motor stops rotating in the forward direction. Then, it sets the RightMotorBackward pin to HIGH, which means the right motor starts rotating in the backward direction. This causes the right wheel to start moving backwards, making the vehicle turn to the right.

Similarly, the function sets the LeftMotorForward pin to HIGH, which means the left motor starts rotating in the forward direction. Then, it sets the LeftMotorBackward pin to LOW, which means the left motor stops rotating in the backward direction. This causes the left wheel to start moving forward, helping the vehicle turn to the right.

### 4.7.4 Training the V3 Module

Training the module is required before using it for voice recognition. Following are the steps to train the module. To connect the circuit to the computer and program the module, follow these steps. First, ensure the circuit is properly connected. Then, launch the Arduino IDE on user's computer. Confirm that the correct Arduino board is selected from the "Tools" menu. Next, verify that the appropriate COM port is chosen under the "Tools" menu. Now, open the sample program for training the module by navigating to "File," then "Examples," and selecting

"VoiceRecognitionV3" followed by "vr_sample_train." Upload the code to the Arduino by pressing Ctrl + U and patiently wait for the upload to complete. Open the Serial Monitor using Ctrl + Shift + M. Ensure the baud rate is set to 115200 and the "Newline" option is selected. If everything is set correctly, the serial monitor will display a menu. To program the module, enter various commands in the serial monitor. For training the module, use the "train" command followed by an address. The syntax for the command is "train address." For example, "train 0," "train 20," or "train 79." In this case, it needs two voice commands to control the LED, one for turning it ON and the other for turning it OFF. Enter the "train" command followed by the desired address, such as "train 20." After entering the command, the serial monitor will prompt you to speak. Speak the command clearly and audibly into the microphone. If the command is recognized properly, the monitor will ask user to speak it again to confirm. If there is any noise during recording or the sound is unclear, it is needed to repeat the command. Keep in mind that the quality of microphone and a noise-free environment plays a role in successful registration of commands.

### 4.7.5 Controlling an LED using voice commands

To control the LED using voice commands, follow these steps. First, open the sample program for LED control by navigating to "File," then "Examples," and selecting "VoiceRecognitionV3" followed by "vr_sample_control_led." In this program, two records are defined: "onrecord" for turning the LED ON and "offrecord" for turning it OFF. Modify the value of "onrecord" to match the address of the voice command that user has trained to turn ON the LED. Similarly, change the value of "offrecord" to match the address of the voice command you trained to turn OFF the LED. Once the modifications are made, upload the code to Arduino. To test the circuit, speak the commands that are trained to turn the LED ON or OFF, just as you programmed them. The Arduino will recognize the voice commands and control the LED accordingly.

### 4.7.6 Gesture Controlled Module Functioning

The ADXL335 3-axis accelerometer module is based on the popular ADXL335 three-axis analog accelerometer IC, which reads off the X, Y, and Z acceleration as analog voltages. By measuring the amount of acceleration due to gravity, an accelerometer can figure out the angle it is tilted at with respect to the earth. By sensing the amount of dynamic acceleration, the accelerometer can find out how fast and in what direction the device is moving. The connections are made such that the X and Y outputs of the accelerometer are connected to the A0 and A1 analog input pins of the Arduino UNO board, respectively. The digital pins D4, D5, D6, and D7 of the Arduino UNO board are connected to the GND (ground) pins of the IN1, IN2, IN3, and IN4 inputs of the relay module, respectively. By connecting the digital pins to the ground pins of the relay module inputs, you are setting them to a low voltage or logic level, effectively turning off the corresponding relay channels. These connections allow the Arduino to read the analog voltage signals produced by the accelerometer in response to acceleration in the X and Y directions.

The mechanism behind controlling the vehicle based on hand movements using the accelerometer acting as a gyro involves several key components. The ADXL335 accelerometer is utilized to detect the hand's movement in different directions such as forward, backward, left, and right. The Arduino UNO board reads the accelerometer data from the X and Y outputs, interpreting the changes in orientation or angular velocity to identify the hand's direction of movement. Based on the detected hand movement, the Arduino triggers specific actions to control the vehicle. For instance, moving the hand forward can signal the Arduino to activate the motor driver or control circuit, causing the vehicle to move forward. Similarly, different hand movements like backward, left, or right can be associated with specific vehicle actions such as reversing or turning in corresponding directions. The motor driver or control circuit receives signals from the Arduino and ensures that the motors operate as intended, enabling the vehicle to respond accordingly. Through this gyroscopic mechanism, the accelerometer and Arduino work in tandem to detect and translate hand movements into control signals, ultimately allowing for gesture-based control of the vehicle.

### 4.7.7 Different movements of vehicle

The code representing stop operation is as follows:

```
if ((xval>294 && xval<340) && (yval>294 && yval<340))

  {
```

```
   digitalWrite(out1,LOW);
   digitalWrite(out2,LOW);
   digitalWrite(out3,LOW);
   digitalWrite(out4,LOW);  }
```

This code is checking if the values of xval and yval variables are within certain range or not. If both variables are within the range of 294 to 340, the code block within the if statement will execute. Inside the if statement, four digital pins (out1, out2, out3, and out4) are set to LOW. This means that these pins will be turned off or set to 0 volts, hence stop operation of vehicle occurs.

The below code controls a motor or device to move forward when the Gyro is in the forward position.

```
else  {
   if ((xval>340 && xval<380) && (yval>294 && yval<340))
  {
    digitalWrite(out1,HIGH);
    digitalWrite(out2,LOW);
    digitalWrite(out3,LOW);
    digitalWrite(out4,LOW);
}
```

The code for backward movement is as follows:

```
   if ((xval>-345 && xval<294) &&  (yval>294 && yval<340))
```

```
 {
  digitalWrite(out1,LOW);
  digitalWrite(out2,HIGH);
  digitalWrite(out3,LOW);
 digitalWrite(out4,LOW);
}
```

The code for left movement is as follows

```
if ((xval>294 && xval<340) && (yval>340 && yval<380))
   {
     digitalWrite(out1,LOW);
     digitalWrite(out2,LOW);
     digitalWrite(out3,HIGH);
     digitalWrite(out4,LOW);
    }
```

The code for right motion of vehicle is as follows

```
if ((xval>294 && xval<340) && (yval>-340 && yval<294))
   {
     digitalWrite(out1,LOW);
```

```
      digitalWrite(out2,LOW);

      digitalWrite(out3,LOW);

      digitalWrite(out4,HIGH);

}
```

### 4.8 Working of ADXL335 Accelerometer

The most used device is the piezoelectric accelerometer. As the name suggests, it uses the principle of piezoelectric effect. The device consists of a piezoelectric quartz crystal on which an accelerative force, whose value is to be measured, is applied.

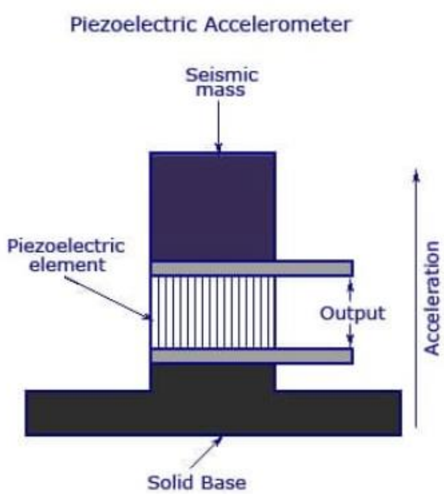


**Figure 4.7 ADXL335 Accelerometer Working**

Due to the special self-generating property, the crystal produces a voltage that is proportional to the accelerative force.

### 4.9 Working of the Project

The system incorporates three modes of operation: Bluetooth, voice control, and gesture control. To facilitate this, the voice and gesture control circuits are integrated onto a single board. The mode selection is achieved through a two-way switch. When the switch is set to 0, the system activates the gyro control application, enabling gesture-based control. Conversely, when the switch is set to 1, the system enters voice command mode, allowing the user to control the vehicle's movements using voice commands. By providing multiple control options and a convenient mode selection switch, the system offers flexibility and versatility in controlling the vehicle.

### 4.9.1 Flowchart of Working

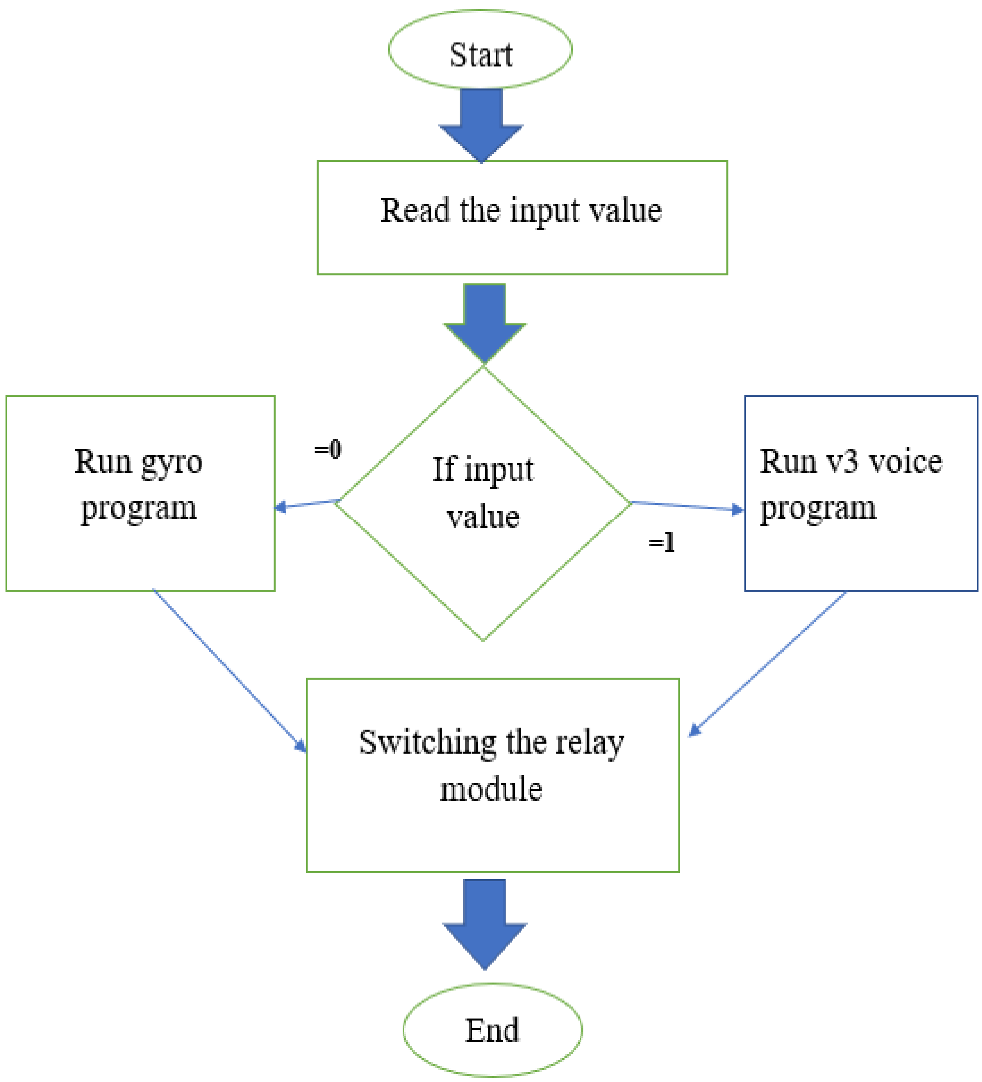


**Figure 4.8 Flow Chart of Control Circuit**

The Algorithm For Above Mention Flow Chart is Depicted in 4.9.2

### 4.9.2 Algorithm Of Entire Working

Step 1: Start

Step 2: Read the input value

Step 3: If a>0 then run the v3 voice program, otherwise run the gyro program.

Step 4: Switching of relays based on gyro or voice inputs.

Step 5: Stop

### 4.10 summary

In this chapter, we have learnt about the functioning of each component which is used in connecting respective hardware components to get the desirable circuit.

# CHAPTER - 5

# SMART CONTROL OF THE VEHICLE USING ARDUINO

## 5.1. Introduction

In this chapter, the overall operation of the developed model including the program dumped in the Arduino Nano is discussed.

## 5.2 Flow chart of voice recognition module

The flowchart for voice control module is depicted in figure 5.1.

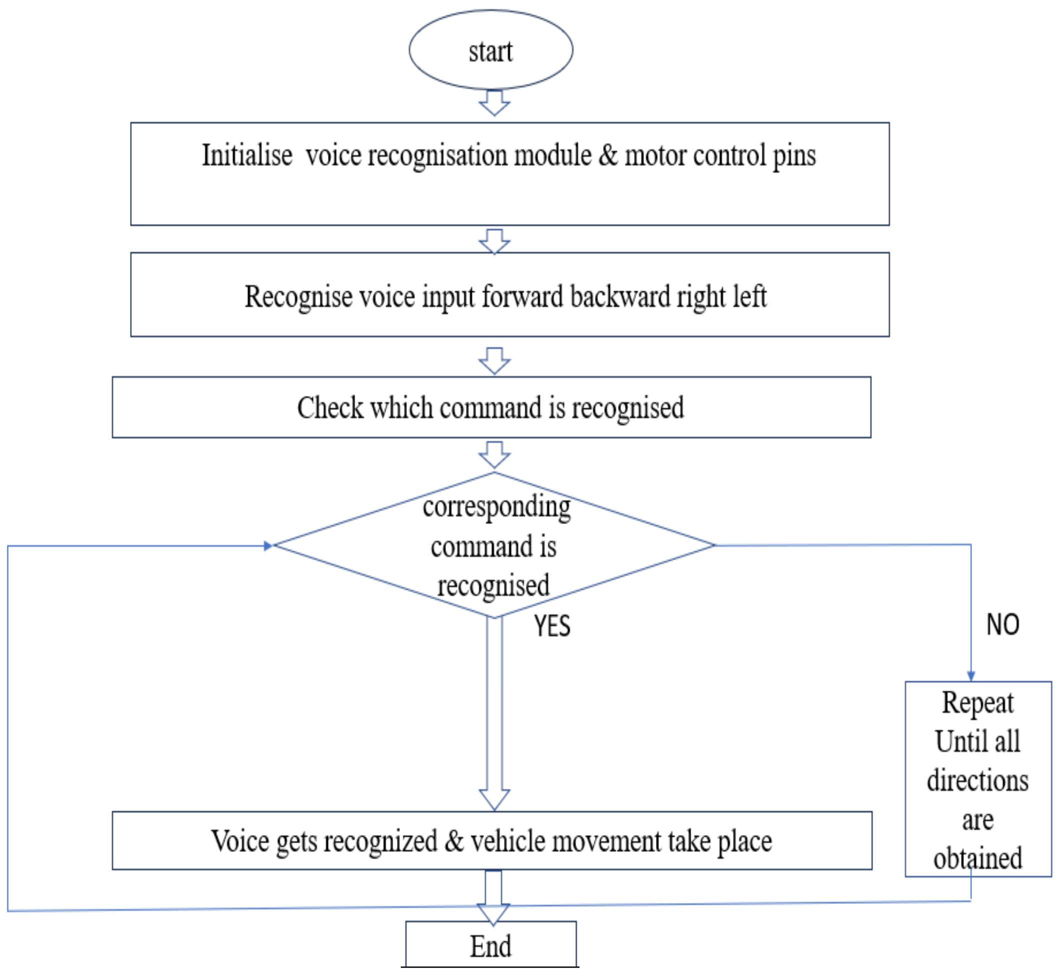


**Figure 5.1 voice control flowchart**

## 5.3 Algorithm of Voice-Controlled Application

- Include required libraries: SoftwareSerial.h and VoiceRecognitionV3.h.
- Create an instance of VR class from VoiceRecognitionV3.h and pass the pins used for communication with the Voice Recognition Module (VRM).
- Define an array named records to store the recorded voice commands and another array named buf to receive the data from the VRM.
- Define pins used to control the motor driver module to control the motors of the car.
- Define constants for the voice commands of the car, such as left, right, forward, backward, and stop.
- Define functions for the various motor movements of the car, such as MotorForward(), MotorBackward(), TurnRight(), and TurnLeft().
- In the setup() function:
- Initialize the VRM and the serial communication with a baud rate of 9600 for the VRM and 115200 for the Arduino.
- Set the defined motor control pins as output.
- Check if the VRM is cleared and load the voice commands for the car.
- In the loop() function:
- Call the recognize() method of the VR class to receive the voice commands from the VRM.
- If a voice command is recognized, execute the corresponding function to control the motors of the car, such as MotorForward() or TurnRight().
- Print the voice command index, group, record number, and signature.
- Call the respective functions for the motor control of the car in the MotorForward(), MotorBackward(), TurnRight(), and `TurnLeft

## 5.4 Programming code for voice-controlled application

```
#include <SoftwareSerial.h>

#include "VoiceRecognitionV3.h"

/**
```

```
  Connection
  Arduino    VoiceRecognitionModule
   2   ------->     TX
   3   ------->     RX
*/
VR myVR(2, 3);
uint8_t records[7];
uint8_t buf[64];
int RightMotorForward = 4;
int RightMotorBackward = 5;
int LeftMotorForward = 6;
int LeftMotorBackward = 7;
void printSignature(uint8_t *buf, int len)
{
  int i;
  for (i = 0; i < len; i++) {
    if (buf[i] > 0x19 && buf[i] < 0x7F) {
      Serial.write(buf[i]);
    }
```

```
    else {
      Serial.print("[");
      Serial.print(buf[i], HEX);
      Serial.print("]");
    }
  }
}
void printVR(uint8_t *buf)
{
  Serial.println("VR Index\tGroup\tRecordNum\tSignature");
 Serial.print(buf[2], DEC);
  Serial.print("\t\t");
  if (buf[0] == 0xFF) {
    Serial.print("NONE");
  }
else if (buf[0] & 0x80) {
    Serial.print("UG ");
    Serial.print(buf[0] & (~0x80), DEC);
  }
```

```
  else {
    Serial.print("SG ");
    Serial.print(buf[0], DEC);
  }
  Serial.print("\t");
Serial.print(buf[1], DEC);
  Serial.print("\t\t");
  if (buf[3] > 0) {
    printSignature(buf + 4, buf[3]);
  }
  else {
    Serial.print("NONE");
  }
  Serial.println("\r\n");
}
void setup()
{
  myVR.begin(9600);
  Serial.begin(115200);
```

```
  Serial.println("Voice Control car");
pinMode(LeftMotorForward,OUTPUT);
pinMode(RightMotorForward,OUTPUT);
pinMode(LeftMotorBackward,OUTPUT);
pinMode(RightMotorBackward,OUTPUT);
  if (myVR.clear() == 0) {
  Serial.println("Recognizer cleared.");
  } else {
    Serial.println("Not find VoiceRecognitionModule.");
    Serial.println("Please check connection and restart Arduino.");
    while (1);
  }
if (myVR.load((uint8_t)left) >= 0) {
    Serial.println("left loaded");
  }
  if (myVR.load((uint8_t)right) >= 0) {
    Serial.println("right loaded");
  }
  if (myVR.load((uint8_t)forword) >= 0) {
```

```
    Serial.println("forword loaded");
  }
  if (myVR.load((uint8_t)backword) >= 0) {
    Serial.println("backword loaded");
  }
}
void loop()
{
  int ret;
  ret = myVR.recognize(buf, 50);
if (ret > 0) {
    switch (buf[1]) {
      case left:
        TurnLeft();
        delay(500);
        MotorStop();
        break;
       case right:
        TurnRight();
```

```
        delay(500);
       MotorStop();
       break;
      case forword:
       MotorForward();
        delay(500);
       MotorStop();
       break;
       case backword:
       MotorBackward();
        delay(500);
        MotorStop();
       break;
default:
       Serial.println("Record function undefined");
       break;
   }
printVR(buf);
  }
```

```
}

void MotorForward() {

  digitalWrite(RightMotorForward, HIGH);

  digitalWrite(RightMotorBackward, LOW);

  digitalWrite(LeftMotorForward, HIGH);

   digitalWrite(LeftMotorBackward, LOW);

}

void MotorBackward() {

    digitalWrite(RightMotorForward, LOW);

  digitalWrite(RightMotorBackward, HIGH);

  digitalWrite(LeftMotorForward, LOW);

   digitalWrite(LeftMotorBackward, HIGH);

}

void TurnRight() {

    digitalWrite(RightMotorForward, LOW);

  digitalWrite(RightMotorBackward, HIGH);

  digitalWrite(LeftMotorForward, HIGH);

   digitalWrite(LeftMotorBackward, LOW);

}
```

```
void TurnLeft() {

  digitalWrite(RightMotorForward, HIGH);

  digitalWrite(RightMotorBackward, LOW);

  digitalWrite(LeftMotorForward, LOW);

   digitalWrite(LeftMotorBackward, HIGH);

}
```

### 5.5 Flow Chart of Gesture-Controlled Module

The below figure 5.2 depicts the flowchart of gesture-controlled module.

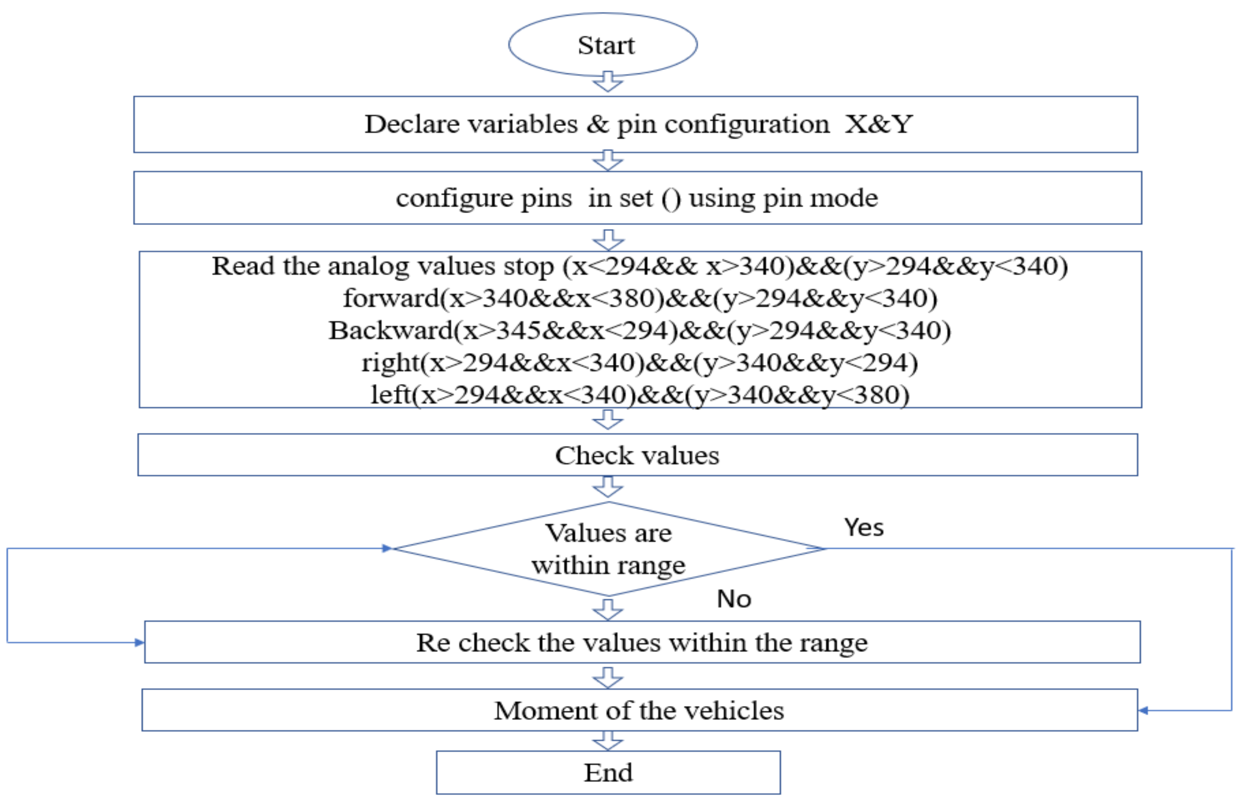


**Figure 5.2 Gesture control flowchart**

### 5.6 Algorithm of Gesture-Controlled Application

- Declare the necessary variables and pin configurations for the input and output pins.
- In the setup() function, configure the input and output pins using pinMode().
- In the loop() function, read the analog values from the x and y pins using analogRead().
- Using a series of conditional statements, check the values of x and y pins to determine the direction of the vehicle.
- For each direction, set the appropriate output pins to either high or low using digitalWrite().
- If the values of x and y pins are within a certain range, set all output pins to low to stop the vehicle.

### 5.7 Programming code for gesture-controlled application

```
int xPin=A0;

int yPin=A1;

int out1=8;

int out2=9;

int out3=10;

int out4=11;

void setup(){

  pinMode(xPin,INPUT);

  pinMode(yPin,INPUT);

  pinMode(out1,OUTPUT);

  pinMode(out2,OUTPUT);
```

```
  pinMode(out3,OUTPUT);

  pinMode(out4,OUTPUT);

}

void loop()

{

  int xval=analogRead(xPin);

  int yval=analogRead(yPin);

if ((xval>294 && xval<340) && (yval>294 && yval<340))

  {

    digitalWrite(out1,LOW);

    digitalWrite(out2,LOW);

    digitalWrite(out3,LOW);

    digitalWrite(out4,LOW);

  }

else  {

    if ((xval>340 && xval<380) && (yval>294 && yval<340))

    {

     digitalWrite(out1,HIGH);

     digitalWrite(out2,LOW);
```

```
     digitalWrite(out3,LOW);
     digitalWrite(out4,LOW);
}
    if ((xval>-345 && xval<294) &&  (yval>294 && yval<340))
  {
   digitalWrite(out1,LOW);
   digitalWrite(out2,HIGH);
   digitalWrite(out3,LOW);
  digitalWrite(out4,LOW);
 }
if ((xval>294 && xval<340) && (yval>340 && yval<380))
    {
      digitalWrite(out1,LOW);
      digitalWrite(out2,LOW);
      digitalWrite(out3,HIGH);
      digitalWrite(out4,LOW);
     }
if ((xval>294 && xval<340) && (yval>-340 && yval<294))
    {
```

```
    digitalWrite(out1,LOW);

    digitalWrite(out2,LOW);

    digitalWrite(out3,LOW);

    digitalWrite(out4,HIGH);

}
```

## 5.8 Control of Motor using Arduino Uno

The operation of the motor is controlled by the user who decides the movement of the motor using a Microphone v3 module or Gyro sensor. A sequential switching operation can be performed based on the usage. By giving requisite commands to the Micro controller via microphone or hand gestures, motor operation is controlled.

The signal is transferred to the Micro Controller via either the microphone or the Gyro sensor. The Micro Controller is connected to the Relay Module, which sends the command to the Arduino Nano for processing.

The rear motor controls the forward and backward motion of the vehicle while the front motor is responsible for left and right motion.

When the input command is received, the Arduino board compares it to the code programmed in it and gives the corresponding command to the driver. This results in the rotation of the motor and movement of the vehicle in the desired direction.

To prevent wear and tear of the equipment and minimize unnecessary friction between the components, any change in motor operation during another operation takes a delay time to proceed to the next operation.

## 5.9 Overall experimental setup

The control circuit of the vehicle can be seen in figure 5.3. Here, in the four relays, each relay corresponds to each operation of the vehicle like forward, backward, left and right.

1. Battery pins
2. Buck converter
3. Motor input pins
4. 4 channel Relay module
5. PWM motor driver
6. Arduino Uno
7. Two way switch

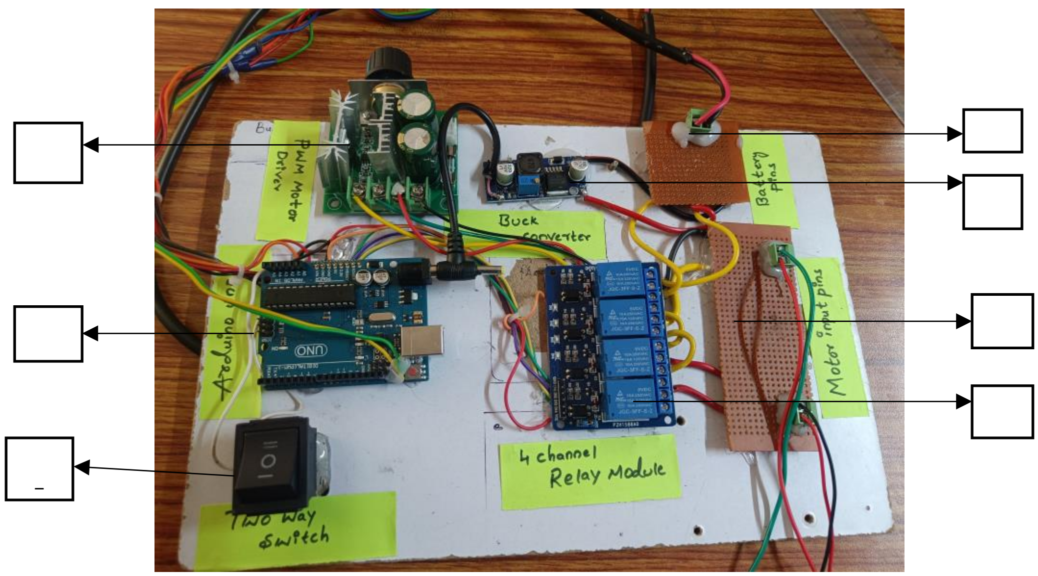


**Figure 5.3 Control Circuit**

The battery connected to the motors and the controlling circuit can be seen in figures 5.2 and 5.3 respectively.

**Figure 5.4 Battery connected to motors.**

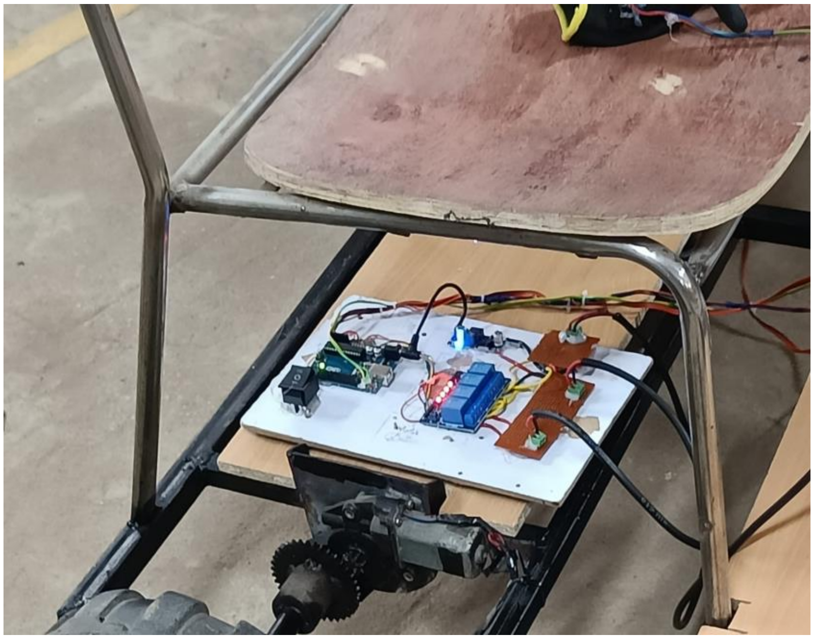

**Figure 5.5 Circuit Setup**

The completed overall model is shown in figure 5.6.

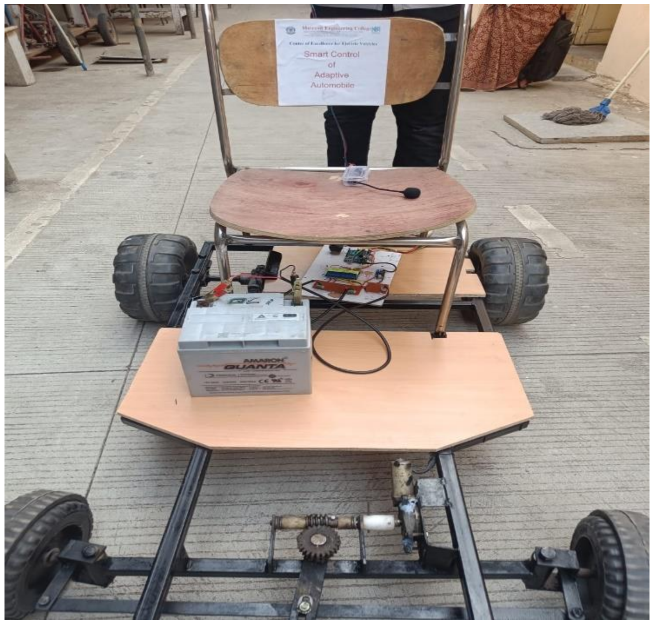


**Figure 5.6 Overall developed model**

### 5.10 Overall Operation of the Developed Model

On supplying current through the model, the vehicle will be ready to receive commands for its operation and so the operator can be seated for his/her travel to another place. The developed model is controlled by Gyro or a microphone. Hence, the next step is to operate the control system of the vehicle by giving voice or gesture commands. This application is built with the help of Arduino Uno. Once the commands are given, the vehicle can perform different movements according to the operator requirements. The commands given to vehicles can be either through voice or gesture and Bluetooth can also be operated with the help of sequential switching operations. The commands given to vehicle are as following:

- Forward
- Backward
- Left
- Right
- Stop

**Figure 5.7 Input command parameters**

The command given by the user /operator through the hand or voice as shown in the figure 5.5 is processed as a signal which is received by the gyro sensor or microphone v3 module, and the signal is then passed to the Arduino then maps the signal with the corresponding command based on the code dumped. The required command is then dispatched to the driver circuit which drives the motor rotation and thus resulting in the movement of the vehicle. To move forward, the operator should say or move their hand towards forward motion similarly, for the backward and stop operation. To take a left-forward turn, first forward and then left commands are to be given similarly, for right-forward. To take a left-backward turn, first forward and then left commands are to be implemented. Similarly, for right backward. A person riding the vehicle can be seen in figure 5.8 below.


**Figure 5.8 Overall Developed Model**

The video captures while testing the project successfully is provided in the link below :

https://youtu.be/9V2MgSyXwyI

## 5.10 Summary

This chapter has dealt with everything about the developed model and its working including with the program dumped in Arduino UNO. Controlling of motor using Arduino is also explained.

## CHAPTER - 6

## CONCLUSION

### 6.1. Conclusion

The voice and gesture control applications for a smart adaptive vehicle has been implemented successfully. By upgrading the existing Bluetooth-controlled vehicle with an embedded system, the project introduced voice and gesture commands as alternative control mechanisms. The integration of a v3 microphone module allowed for the recording of voice commands, while a gyroscope enabled gesture-based input for individuals who cannot speak.

The implemented vehicle demonstrated various operations, including forward, backward, left, right, wake-up, and shut-down. Sequential operations were facilitated through the addition of a switching module, ensuring smooth transitions between different commands.

Overall, the project successfully enhanced the connection between humans and robots by providing an accessible and user-friendly control system for the smart adaptive vehicle. The integration of voice and gesture control applications improved the mobility and independence of blind or hand-disabled individuals, enabling them to navigate small distances with ease.

**6.1.Future scope**

In the future, voice recognition features can be enhanced by the latest technologies like Alexa, Internet of things, which can be controlled by mobile phone. The speed regulation and four wheeled drive etc. other kinds of developments can be used to make the project available to all people in need and meet all desirable needs in using this vehicle.

## REFERENCES

The journals used for reference are:


1. “Hand Gesture Based Wheelchair Movement Control for Disabled Person Using MEMS”, Prof. Vishal V. Pande, Nikita S.Ubale, Darshana P. Masurkar, Nikita R. Ingole, Pragati P. Mane, International. Journal of Engineering Research and Applications, vol. 4, no. 4, pp. 152-158, 2014, ISSN 2248-9622.
2. "Hand Gestures Controlled Wheel Chair", Mahipal Manda, B Shankar Babu, M Abhishek and J Srikanth, International Journal of Emerging Technologies and Innovative Research, vol. 5, no. 9, pp. 642-647, 2018, ISSN 2349-5162.
3. "Gesture controlled wheelchair using IR-LED T SOP pairs along with collision avoidance", Rajesh Kannan Megalingam, Clarion Chacko, Binu P. Kumar, Ashwin G. Jacob and P. Gautham, International Conference on Robotics and Automation for Humanitarian Applications (RAHA), 2016.
4. "Wireless Gesture Control Wheelchair", Shanelle Fernandes, Rushia Fernandes and Jessica Kakkanad, International Journal of Recent Technology and Engineering (IJRTE), vol. 9, no. 2, July 2020, ISSN 2277-3878.
5. "Automated Gesture Based Wireless Wheelchair Control by Means of Accelerometer”, V Sridevi, P. Ishwarya, P. Surya Chandra and N. Suresh Kumar, International Journal of Engineering and Advanced Technology (IJEAT), vol. 9, no. 1, October 2019, ISSN 2249-8958.
6. "Gesture Controlled Mobile Robotic Arm for Elderly and Wheelchair People Assistance Using Kinect Sensor", M. Ababneh et al, 2018 15th International Multi-Conference on Systems Signals & Devices (SSD), 2018.
7. "A hand gesture based wheelchair for physically handicapped person with emergency alert system", P.P. Chitte, S.B. Khemnar, A.A. Kanawade and S.B Wakale, International Research Journal of Engineering and Technology (IRJET), vol. 03, no. 04, Apr 2016.
8. "A Smart-hand Movement-based System to Control a Wheelchair Wirelessly", Fahd N. Al-Wesabi, Mohammad Alamgeer, Fuad Al-Yarimi and Adnan Albaadani, Sensors and Materials, vol. 31, no. 9, pp. 2947-2964, 2019.